

Aleksi Rajakoski

# Ruby on Rails -pohjainen asiakas- portaali

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tietotekniikka

Insinöörityö

30.4.2014

Tekijä	Aleksi Rajakoski
Otsikko	Ruby on Rails -pohjainen asiakasportaali
Sivumäärä	34 sivua
Aika	30. huhtikuuta 2014
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Ohjelmistotekniikka
Ohjaajat	Lehtori Simo Silander Johtava konsultti Aki Teronen
<p>Tässä insinöörityössä perehdyttiin Fluido Oy:n asiakkaiden käyttöön tulevan Ruby on Rails -pohjaisen Fluido Portal -sovelluksen luomiseen Heroku-pilvipalvelualustalle, jota käyttäjät voivat hyödyntää oman Salesforce.com-instanssinsa kautta kolmannen osapuolen Force.com Canvas -ohjelmalla.</p> <p>Työssä käytiin myös tiivistettynä läpi Ruby on Rails -pohjaisen Fluido Portal -sovelluksen rakennetta ja siihen liittyviä teknologioita sekä esiteltiin lyhyesti Heroku- ja Salesforce.com-pilvipalvelualustoja.</p> <p>Teknologiaosuuden jälkeen esiteltiin Fluido Portalin ja Salesforce.comin yhdessä muodostamaa tietoturvaa, jossa hyödynnetään Force.com Canvas -rajapintaa.</p> <p>Fluido Portalin tietokantamalli -osuudessa esiteltiin käytetyn PostgreSQL-tietokannan taulujen rakenne, sekä käytiin lyhyesti läpi Ruby-ohjelmointikielen Active Directory -toiminnallisuus ja objektien välisten assosiaatioiden luominen.</p> <p>Loppupäätelmissä todettiin, ettei toteutettu sovellus ole vielä julkaisukelpoinen, ja sille on jatkokehityssuunnitelmia. Myös Herokun maksullisten ominaisuuksien käyttöönottoa suositeltiin, jotta asiakkaille voitaisiin tarjota sulava käyttökokemus sovellusta ladattaessa.</p>	
Avainsanat	Heroku, Salesforce.com, Portaali, Ruby on Rails, Pilvipalvelut, PostgreSQL, Force.com Canvas

Authors	Aleksi Rajakoski
Title	Ruby on Rails -based Customer Portal
Number of Pages	34 pages
Date	30. huhtikuuta 2014
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructors	Simo Silander, Senior Lecturer Aki Teronen, Head of Development
<p>The aim of this bachelor's thesis was to focus on the development process of an upcoming Ruby On Rails -based Fluidio Portal -application that will be available for Fluidio Ltd's customers. The application itself will run on Heroku's cloud service platform. Users will be able to utilize Fluidio Portal through their own Salesforce.com instances as a third-party Force.com Canvas -application.</p> <p>This study also briefly explains the structure of the Ruby on Rails -based Fluidio Portal and the technologies used in it. Also Heroku and Salesforce.com cloud service platforms are briefly introduced.</p> <p>After the technology part, the combined security of Fluidio Portal and Salesforce.com is explained, as well as the use of Force.com Canvas API in that process.</p> <p>The part on the data model of Fluidio Portal explains the table structures in PostgreSQL database used in the application. It also contains a brief review of the Active Record functionality of the Ruby programming language and the process of creating associations between objects.</p> <p>The developed application did not reach its final version, and further development is needed. The use of Heroku's non-free features with Fluidio Portal is recommended based on the study since it would provide the customers with a better user experience.</p>	
Keywords	Heroku, Salesforce.com, Portal, Ruby on Rails, Cloud Services, PostgreSQL, Force.com Canvas

# Sisältö

1	Johdanto	1
2	Fluido Portalin suunnittelu ja vaatimukset	3
2.1	Idea ja toiminnallisuuksien määrittäminen	3
2.2	Kehitysympäristöt ja Fluido Portalin luominen	4
2.3	Käyttöliittymän suunnittelu	4
3	Fluido Portalissa käytettävät teknologiat	6
3.1	Salesforce.com	6
3.2	Ruby-ohjelmointikieli	7
3.3	RVM-versionhallinta ja Gem-paketit	7
3.4	Ruby on Rails -ohjelmistokehys	8
3.5	PostgreSQL-tietokantajärjestelmä	9
3.6	Bootstrap-ohjelmistokehys	9
3.7	Heroku-pilvipalvelualusta	10
3.8	Force.com Canvas -ohjelmistokehys	11
3.9	GIT-versionhallinta	12
4	Autentikointi ja tietoturva	13
4.1	Autentikoinnin ja tietoturvan yleiskuvaus Fluido Portalissa	13
4.2	Force.com Canvas -rajapinnan autentikaatio	14
4.3	Salesforce.com-puolen käyttöoikeuksien hallinta	15
5	Fluido Portalin tietokantamalli	16
5.1	Tietorakenne	16
5.2	Tietokannan migraatitiedostot	17
5.3	Assosiaatioiden luominen ja Active Record	19
5.4	Datan synkronointi Salesforce.com-instanssin kanssa	20
6	Fluido Portalin kehitysversio ja jatkokehityssuunnitelmat	23

6.1	Sovelluksen arkkitehtuurikuvaus	23
6.2	Sovelluksen tiedosto- ja kansiorakenne	25
6.3	Versionhallinnassa käytetty puurakenne	26
6.4	Käyttöliittymän esittely	28
6.5	Jatkokehityssuunnitelmat	30
7	Yhteenveto ja päätelmät	32
	Lähteet	34

## Käsitteet

**CRM** Customer Relationship Management, eli asiakkuudenhallinta.

**CSS** Cascading Style Sheet. Sisältää HTML-pohjaisten näkymien ulkoasumääritteet.

**Dyno** Herokun tarjoama virtuaalinen laskentayksikkö ajettaville sovelluksille.

**Gem** Ruby-kehittämistä helpottava valmis laajennus/toiminnallisuus.

**Hiekkalaatikko** Kehitys-/testiympäristö ohjelmistoille ja järjestelmille.

**HTML** Hypertext Markup Language. Muun muassa verkkosivujen kehittämisessä käytetty kuvauskieli.

**Ohjelmistokehys** Muodostaa rungon sen päälle rakennettaville ohjelmistoille.

**Pilvialusta** Verkkopohjainen alusta toiminnallisuuksille.

**Portaali** Verkkosivustopohjainen palvelu, joka tarjoaa pääsyn muihin Internet-palveluihin.

**REST** Representational State Transfer. HTTP-protokollaan perustuva arkkitehtuurimalli ohjelmistorajapinnoille.

**Salesforce.com** SaaS-pohjainen pilvipalvelu / Yhdysvaltalainen yritys.

**Tietokanta** Tietosäiliö, jota voidaan hyödyntää esimerkiksi verkkosivuilla tai muissa ohjelmissa.

**Tuotantoympäristö** Ohjelmistojen toimintaympäristö, joka on virallisesti käytössä, sisältäen julkaistuja toiminnallisuuksia.

**Versionhallinta** Ohjelmistokehityksen työkalu, joka helpottaa lähdekoodien hallintaa.

**Visualforce** Salesforce.com:ssa käytetty HTML-pohjainen verkkosivustonäkymä.

## 1 Johdanto

Tässä insinööriyössä esitellään Ruby on Rails -pohjaisen *Fluido Portal* -sovelluksen suunnitteluvaiheita, siinä käytettäviä teknologioita, kehitystyötä ja rakennetta. Työ tehdään Fluido Oy:lle, joka on vuonna 2009 perustettu Salesforce.com-pilvipalveluun erikoistunut pohjoismainen konsulttitalo, joka tarjoaa asiakkailleen räätälöityjä konsultointi- ja kehityspalveluita sekä koulutuksia.

Fluido Portal -sovelluksen tavoitteena on tarjota Fluido Oy:n asiakkaille mahdollisuus seurata heidän projektiensa etenemistä. Tämän lisäksi sen avulla pyritään helpottamaan laskutettujen tuntimerkintöjen selaamista tarjoamalla asiakkaille suora pääsy seuraamaan käytettyjen tuntien sisältämää tehtyä työtä. Asiakkaiden pääasiallinen työnteko tapahtuu verkkosivustopohjaisessa Salesforce.comin CRM-järjestelmässä, jonka vuoksi käyttäjien ohjaaminen erilliselle verkkosivustolle ei olisi luontevaa. Tämän vuoksi Fluido Portalin käyttökanavana toimii Salesforce.com-ympäristön tarjoama Canvas App -ohjelmistorajapinta, jonka avulla sovellus voidaan näyttää *Visualforce*-sivulla Salesforce.com-alustan sisällä. Näin ollen asiakkaan ei tarvitse siirtyä pois senhetkiseltä sivustoltaan selatakseen esimerkiksi omien projektiensa tietoja, eikä heidän tarvitse erikseen autentikoitua sovellukseen sen käyttöä varten. Fluido Portalia käytettäessä asiakkaat voivat nähdä vain heidän omiin projekteihinsa liittyvää dataa.

Koska portaali on rakennettu erillisenä sovelluksena Herokun tarjoamalle pilvipalvelualueelle, eikä suoraan Fluidon omaan Salesforce.com-instanssiin, pystytään myös säästämään maksullisia Salesforce.com-käyttäjälisenssejä, joita muuten olisi tarvittu asiakkaiden käyttöön autentikoinnissa. Samalla sovellusta voidaan jatkossa esitellä asiakkaille yhtenä ratkaisuna heidän haasteisiinsa, joissa jokin ulkoinen järjestelmä täytyisi saada keskitetysti käyttöön myös Salesforce.comin puolella.

Fluido Portalin käyttöä varten asiakkaan Salesforce.com-tuotantoympäristöön asennetaan Fluidon hallinnoima sovelluspaketti, jonka avulla Fluido Portaliin pääsevät käsiksi ne käyttäjät, joille sen käyttöön vaadittavat oikeudet on jaettu. Asiakas pystyy itse määrittämään Salesforce.comin sisäisellä oikeuksien hallinnalla, ketkä sovellusta voivat käyttää.

Fluido Portalin käyttämän Heroku-pilvipalvelussa sijaitsevan PostgreSQL-tietokannan sisältämä data haetaan Fluidon omasta Salesforce.com-tuotantoympäristöstä. Tietokantojen välinen tarvittavan datan synkronointi toteutetaan automaattisesti, tai tarvittaessa myös manuaalisesti käyttäen siihen erikseen kehitettyä erillistä sovellusta. Käytettävä synkronointisovellus jakaa Fluido Portal -sovelluksen kanssa saman tietokannan, mutta toimii eriytettyinä pääsovelluksesta, jotta sen tarjoaman toiminnallisuuden vaihdettavuus toiseen sovellukseen olisi siistimpää ja helpompaa.

Työn lopuksi pohditaan Herokun maksullisten ominaisuuksien käyttöönottamisen tarpeellisuutta ja niiden tarjoamia etuja, sekä esitellään Fluido Portalin mahdollisesti toteutettavia jatkokehityssuunnitelmia.



## 2 Fluido Portalin suunnittelu ja vaatimukset

### 2.1 Idea ja toiminnallisuuksien määrittäminen

Fluido Portal -sovelluksen kehityksen takana oli tarve toteuttaa asiakkaille yksinkertainen tapa selata heidän Fluidolle ulkoistamia projekteja ja niiden laskutettujen tuntien kirjauksia. Tämän lisäksi sen tulisi toimia pohjana, jonka päälle voidaan tulevaisuudessa laajentaa muitakin toiminnallisuuksia. Fluido Portalin tulisi myös toimia yleisenä esimerkkinä Salesforce.com-ympäristön sisällä *iFramessa* näytettävien Force.com Canvasta hyödyntävien sovelluksien toiminnasta. Tällöin sitä voitaisiin esitellä asiakkaille mahdollisena ratkaisuna heidän tarpeisiinsa, jossa jokin ulkoinen sovellus täytyy saada sisällytettyä Salesforce.comiin.

Määriteltäessä Fluido Portalin toiminnallisia vaatimuksia, täytyi ottaa huomioon erilaisen toteutusmahdollisuuksien hyödyt ja heikkoudet, joiden analysoinnin avulla voitiin valita paras vaihtoehto. Yhtenä vaatimuksena sovellukselle päätettiin, että se täytyi saada toimimaan Heroku-pilvipalvelualustan päällä, joka tarjoaa erilaisille verkkopohjaisille sovelluksille alustan niiden suorittamista varten. Toteutuksen olisi voinut kehittää myös Salesforce.com-alustan sisäisesti, mutta kyseinen ratkaisu olisi syönyt muun muassa turhaan maksullisia Salesforce.com-lisenssejä, joita oltaisiin tarvittu käyttäjien autentikointimista varten. Toinen vaatimus oli saada yksinkertainen autentikointitapa, jossa käyttäjän ei tarvitsisi omistaa erillisiä tunnuksia Fluido Portalia varten, vaan pystyisi käyttämään jo olemassa olevia Salesforce.com-tunnuksiaan. Kolmantena vaatimuksena oli selkeä ja helppo pääsy Fluido Portaliin.

Fluido Portalin osalta päädyttiin kehittämään Ruby on Rails -pohjaista sovellusta, joka käyttää hyväkseen Force.com Canvas -rajapintaa, jonka avulla kolmannen osapuolen ulkoisia sovelluksia voidaan ajaa Salesforce.com-ympäristössä vuoravaikutteisuuden säilyessä niiden välillä. Tämän olisi tarkoitus mahdollistaa yksinkertainen toteutus sekä helppo käyttöisyys asiakkaille, jonka oletettiin vaikuttavan suoraan Fluido Portalin käyttöasteeseen.

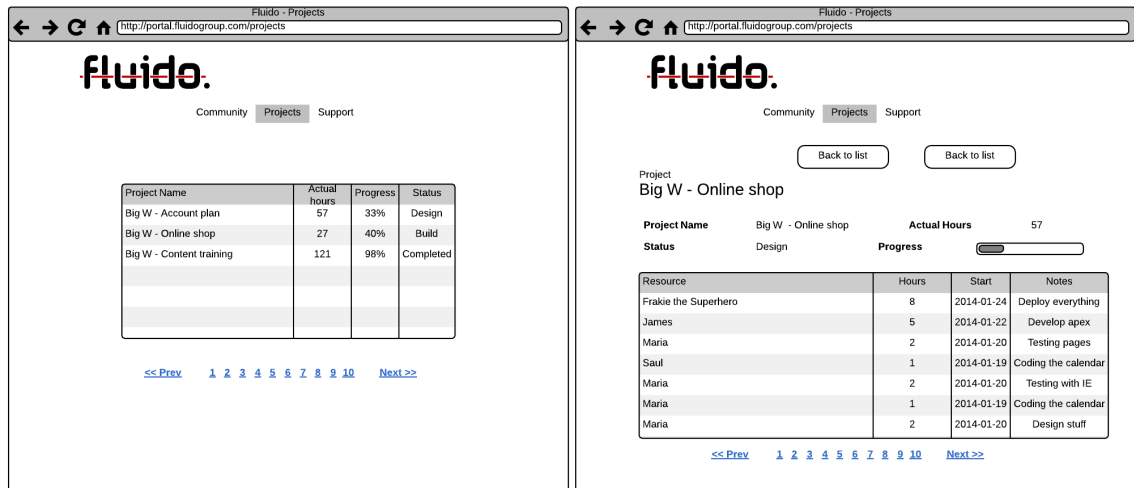
## 2.2 Kehitysympäristöt ja Fluido Portalin luominen

Fluido Portalille asetettujen vaatimusten määrittämisen jälkeen oli mahdollista lähteä miettimään toteutuksen tarkempia tapoja ja käytettäviä teknologioita. Tämän projektin toteuttamiseksi kehitysympäristöinä toimivat vapaan lähdekoodin Linux (Elementary OS) sekä Applen OS X Mavericks -käyttöjärjestelmät, joissa molemmissa pyörii paikallinen PostgreSQL-tietokanta. Tämän lisäksi niihin on asennettu Rubyn kehitystyökalut, Heroku Toolbelt, Git-työkalut ja jokin ohjelmointikielet tunnistava tekstieditori. Valitut käyttöjärjestelmät tarjoavat hyvän tuen Ruby-kehittämiselle, ja niille on myös tarjolla laaja valikoima hyvin toimivia sovelluksia, joita voidaan tarvittaessa ottaa käyttöön. Kehitysympäristön pohjana toimivan käyttöjärjestelmän kohdalla ei kuitenkaan ole erityisiä rajoituksia, sillä lähes kaikki yleisimmät käyttöjärjestelmät sisältävät tuen Ruby on Rails -kehittämiselle, Herokun työkaluille ja Git-työkaluille.

Fluido Portalin kehitystyön aluksi oli helpointa luoda sitä varten uusi tyhjä ohjelma Herokuun, jonka sovellusvarasto kloonataan Gitin avulla paikalliseen tiedostojärjestelmään. Tämän jälkeen Ruby on Rails -ohjelman pohja voidaan luoda käyttäen komentorivillä komentoa `'rails new .'`, joka lisää projektikansioon sovelluksen tarvitsemat tiedostot ja kansiorakenteen. Kehitystyötä tehtäessä Fluido Portalia on ajettu paikallisesti Ruby on Rails -Gemin tarjoaman `'rails server'` -komennon avulla, jota käyttämällä sovelluksen HTML-pohjaisia näkymiä ja toimintalogiikkaa voidaan kokeilla valituilla verkkoselaimilla. Ohjelmaa voidaan myös tarvittaessa rajoitetusti muokata ajonaikaisesti lennossa, ilman sovelluksen ajamista väliaikaisesti alas. Kun jokin tarvittava ominaisuus valmistuu, sovelluksen uusi versio voidaan siirtää versionhallintatyökalun (Git) avulla takaisin Herokuun, ja yhdistää se myös muihin olemassa oleviin versionhallinnan kehityshaaroihin.

## 2.3 Käyttöliittymän suunnittelu

Fluido Portal -projektin alkuvaiheilla graafisesta käyttöliittymästä toteutettiin alustavia rautalankamalleja (*wireframes*), jotka antavat suuntaa käyttöliittymän ja toiminnallisuuksien kehittämiselle.



Kuva 1: Rautalankamallit projektien listauksen ja projektin tarkastelun näkymistä.

Fluido Portalia varten *LucidChart*-ohjelmalla nopeasti toteutetut rautalankamallit (kuva 1) projektilistauksesta ja projektin tietojen tarkastelunäkymästä toimivat pohjana kehitystyön aloittamiselle. Rautalankamalleista pystyy tässä tapauksessa päättämään, mitä dataa eri näkymissä voitaisiin näyttää ja miten. Tämän lisäksi sovelluksen logiikkaa ohjaimissa ja malleissa voidaan kehittää oikeaan suuntaan, jotta tarvittavat datat ja toiminnallisuuksien liipasimet ovat eri näkymissä käytettävissä. Fluido Portalin rautalankamallit ovat hyvin yksinkertaiset, mutta joskus ohjelmisto/verkkosivustoprojektien käyttöliittymäsuunnittelussa mukana olevat graafiset suunnittelijat ovat saattaneet luoda toteutuksesta hyvinkin tarkan esimerkin. Niissä grafiikat, asetellut ja tekstit ovat ennestään määritettyinä, jolloin sen toteuttaminen on huomattavasti työläämpää, sillä lopputuloksen osalta ei ole yhtä paljon vapauksia.

### 3 Fluido Portalissa käytettävät teknologiat

#### 3.1 Salesforce.com

Tähän työhön olennaisesti liittyvä Salesforce.com on SaaS (Software as a Service) -tyyppinen pilvipalvelu, joka toimii CRM (Customer Relationship Management) -tuotteena. Sen takana on vuonna 1999 Yhdysvalloissa perustettu Salesforce.com-yhtiö, jonka liikevaihto vuonna 2013 oli 3 miljardia Yhdysvaltain dollaria. Se on myös Forbesin toimesta valittu innovatiivisimmaksi yhtiöksi Yhdysvalloissa. Tekniseltä puoleltaan Salesforce.com käyttää hyväkseen Oraclen tarjoamia teknologioita kuten Oracle Linuxia, Oracle Exadataa, Oracle Databasea ja Java-alustaa ohjelmiensa ja SaaS-alustansa pohjana. [1]

Salesforce.com tarjoaa CRM-ratkaisun asiakkailleen. Sen käyttö skaalautuu isojen ja pienten yritysten käyttöön, jotka haluavat säilyttää järjestelmässään tietoja esimerkiksi asiakkaistaan, tuotteistaan, myynneistään ja muista haluamistaan asioista. Järjestelmä tarjoaa monipuolisia vaihtoehtoja sen kustomoimiseen, kuten avoimen ohjelmointirajapinnan ja joukon visuaalisia työkaluja. [2] Näiden avulla asiakas voi muokata alustan vastaamaan paremmin omia tarpeitaan, esimerkiksi lisäämällä omia tietueita ja automatisoimalla yleisiä toimintojaan tai rakentamalla täysin omia toiminnallisuuksia ja verkkosivuja. Järjestelmään voidaan myös asentaa erilaisia kolmannen osapuolen kehittämiä ohjelmia, joilla toiminnallisuuksia pystytään laajentamaan. Yksi kustomointiin liittyvistä hyödynnettävistä ominaisuuksista on Force.com Canvas -rajapinta, jota myös Fluido Portal hyödyntää. Ohjelmointikielenä Salesforce.comissa käytetään Apexia, jonka avulla luodaan muun muassa kustomoituja liipaisimia tietueiden tietokantaoperaatioiden yhteyteen ja ohjainluokkia Visualforce-pohjaisiin verkkosivunäkymiin.

Salesforce.comin suosio Pohjoismaissa on alkanut vasta viimeisten vuosien aikana kasvaamaan reilusti, ja on samalla alkanut ottaa jalansijaa yrityksissä. Yhdysvalloissa sen suosio on ollut jo pidemmän aikaa erittäin korkealla. Pahin kilpailija Salesforce.comille on Microsoftin tarjoama *Dynamics CRM*.

### 3.2 Ruby-ohjelmointikieli

Yukihiro Matsumoton vuonna 1995 ensimmäistä kertaa julkaisema Ruby-ohjelmointikieli on dynaaminen oliopohjainen ohjelmointikieli, jonka vaikutteina ovat toimineet muut ohjelmointikielet kuten Perl, Smalltalk, Eiffel, Ada ja Lisp. Se tukee proseduraalisia ja funktionaalisia ohjelmointityylejä ja on puhtaasti olio-pohjainen, jolloin kaikki elementit ovat objekteja. Ruby tukee dynaamista muuttujien tyyppitystä, joita se myös tarkistaa ajonaikaisesti, sekä ankka-tyypitystä. Tämän avulla kehittäjän ei tarvitse erikseen määrittää muuttujan tyyppiä esimerkiksi numeeriseksi, vaan sen tyyppi määräytyy sen mukaan, millainen arvo siihen asetetaan. Matsumoton ideana oli kehittää ohjelmointikieli, jossa panostetaan inhimillisyyteen, hauskuuteen ja ohjelmoijan tuottavuuden tehostamiseen. Sen sisäinen toiminta on monimutkaista, mutta sen syntaksi on pyritty säilyttämään yksinkertaisena kehittäjille. [3] Fluido Portalissa käytettävä Ruby-versio on 2.1.0, joka julkaistiin virallisesti vuoden 2013 loppupuolella.

### 3.3 RVM-versionhallinta ja Gem-paketit

Ruby-kehitysympäristöön liittyy olennaisesti RVM (Ruby Version Manager), joka nimensä mukaisesti toimii Rubyn versionhallintana. Koska Ruby on jatkuvasti kehitystyön alla oleva ohjelmointikieli, sen ominaisuudet saattavat muuttua merkittävästi eri versioiden myötä. Muutokset puolestaan saattavat johtaa eri versioilla kehitettyjen projektien rikoutumiseen, jonka vuoksi niihin kannattaa määritellä käytettävän Ruby-version numero. RVM mahdollistaa eri versioiden asentamisen helposti, joten tiettyä projektia on helppo kehittää oikealla Ruby-versiolla. Jos sovelluksen kehitystyön aikana sille ei kuitenkaan ole määritetty erikseen käytettävän Rubyn versionumeroa, sovellusta suorittava alusta, kuten Heroku, määrittää sille automaattisesti jonkin käytettävän versionumeron. Tämä voi johtaa sovelluksen virheelliseen toimintaan ja pahimmillaan jopa täydelliseen toimimattomuuteen.

Kehitysvaiheessa Ruby-ohjelmia voidaan laajentaa käyttämällä valmiita toiminnallisuksia sisältäviä Gem-paketteja, joilla voidaan esimerkiksi ottaa käyttöön jokin tietokantajärjestelmä tai verkkosivustolaajennus. Gemejä asennetaan ja päivitetään RubyGems-pakettienhallinnalla. Gemit voivat olla joko pieniä toiminnallisuuksia sisältäviä paketteja,

tai erittäin laajoja pakettikokonaisuuksia, kuten Ruby on Rails. Jos siis jokin sopiva toiminnallisuus on jo ennestään kehitetty ja julkaisu Geminä, on yleensä turhaa lähteä kehittämään vastaavaa toimintoa itse. Gemejä voidaan osittain verrata esimerkiksi muissa kielissä käytettäviin kirjastoihin ja niiden tarjoamiin funktioihin. Gemejä hyödyntämällä voidaan siis säästää huomattavasti aikaa sovelluksia kehitettäessä. Käyttämällä *Bundler*-toiminnallisuutta sovelluksessa käytettäviä Gemejä voidaan hallinnoida lisäämällä niitä tiedostoon, joka sisältää listauksen kaikista halutuista Gemeistä ja valinnaisesti niiden versioista. Gemit voidaan tämän jälkeen asentaa sovelluksen käytettäväksi sen kansioon komennolla *'bundle install'*.

Gem-laajennuksia voidaan myös kehittäjien toimesta tarvittaessa toteuttaa lisää. Fluido Portal -sovelluksen tapauksessa kehitystyön tuloksena toinen sovelluksen kehittäjistä toteutti uuden erillisen Gem-laajennuksen, jota voidaan jatkossa hyödyntää mahdollisten uusien Ruby on Rails -sovellusten yhteydessä. Kehitetty Gem toteuttaa Force.com Canvas-toiminnallisuuden vaatiman autentikoinnin kehitettävän sovelluksen ja Salesforce.comin välillä, samalla tarjoten myös apufunktiota Force.com Canvasin käyttöön liittyen. Gemiä ei ole julkaistu vapaasti muiden käyttöön, vaan se säilytetään Fluido Oy:n yksityisessä omistuksessa.

### 3.4 Ruby on Rails -ohjelmistokehys

Geminä asennettava Ruby on Rails (*RoR*, *Rails*) on vuonna 2005 julkaistu Rubyn päälle kehitetty avoimen lähdekoodin ohjelmistokehys (*framework*), joka pohjautuu MVC-arkkitehtuuriin. Railsia käytetään lähinnä WWW-pohjaisten sovellusten kehittämiseen. [4] MVC-arkkitehtuurin mukaiset mallit (*model*) luodaan Ruby-luokkina, jotka vastaavat joitain käytettäviä objekteja. Myös tietokannan tauluja voidaan käyttää objekteina, jolloin niiden sisältämät kentät käännetään luokkamuuttujiksi. Sovelluksen näkymät, eli käyttöliittymä, toteutetaan HTML-pohjaisina tiedostoina, joissa haluttu data ja erilaisia toiminnallisuuksia laukaisevat painikkeet näytetään. Niiden esittämä data ja toiminnallisuudet saadaan käyttämällä sen ohjainta (*controller*), joka on toteutettu myös omalla Ruby-luokkanaan. Tarkempi kuvaus Ruby on Rails -sovellusten rakenteesta esitellään myöhemmin tämän työn luvussa 6.

Ruby on Rails sisältää myös monia sisäänrakennettuja työkaluja, joilla voidaan toteuttaa erilaisia kehittäjiä helpottavia asioita. Yksi eniten käytettävistä toiminnallisuuksista on komennon *'rails server'* avulla luotavan paikallisen Ruby on Rails -palvelimen käynnistäminen, jotta sovellusta voidaan kokeilla paikallisesti verkkoselaimella. Toinen merkittävistä toiminnallisuuksista on *generate*-toiminnallisuus, jolla automaattinen yksittäisten luokkien, näkymien ja muiden tarvittavien tiedostojen luominen on mahdollista. Komennolla *'rails generate scaffold [jokin nimi]'* voidaan tarvittaessa myös luoda yhdellä kertaa yksittäistä asiaa varten kokonaisuuksia, jolloin näkymät, mallit, ohjaimet, tarvittavat tietokannan migraatiotiedostot, sekä muut tiedostot luodaan automaattisesti. Tämä ei kuitenkaan aina ole suositeltava käytäntö, sillä turhien tiedostojen luonti voi aiheuttaa jatkossa ongelmia, eikä kehittäjä välttämättä muista, mitä kaikkea takana on tapahtunut.

### 3.5 PostgreSQL-tietokantajärjestelmä

Avoimeen lähdekoodiin perustuva PostgreSQL on olio-relaatiotietokantapalvelin, joka tarjoaa vaihtoehdon muille tietokantajärjestelmille, joita ovat MySQL, Oracle tai Microsoft SQL Server. PostgreSQL on hieman hitaampi tietokantajärjestelmä kuin MySQL, mutta sen ominaisuudet ja luotettavuus todetaan vertailussa paremmiksi. Se myös tukee kaikkia tyypillisiä ACID-ominaisuuksia, kuten transaktioita ja viittauseheyden tarkistusta. PostgreSQL:n versio 1 julkaistiin vuonna 1989. [5] Fluido Portalin paikallisesti kehitettävässä versiossa käytetään PostgreSQL:n versiota 9.3, joka pohjautuu vuonna 2010 julkaistuu 9.0-versioon.

### 3.6 Bootstrap-ohjelmistokehys

Verkkosivustoilla käytettävät tyyliohjelmat saattavat olla työläitä ja vaikeasti ylläpidettäviä, mikä johtuu jatkuvasti kehittyvistä selaimista ja niiden yhteensopivuusongelmista. Alun perin Twitterin työntekijöiden aloittama Bootstrap-projekti tähtää ylläpidettävyyden helpottamiseen. Se julkaistiin vuonna 2011 vapaan lähdekoodin projektina julkiseen käyttöön GitHubissa, ja oli seuraavana vuonna sen suosituin kehitysprojekti. [6] Bootstrapin käyttö verkkosivujen pohjana on nykyään erittäin yleistä, johtuen juuri ylläpidettävyyden ja sen käytön helppoudesta. Sen avulla voidaan luoda nopeasti verkkosivuja, jotka näyttävät hyvältä ja ovat pitkälti yhteensopivia eri verkkoselaimissa ja niiden versioissa. Myös Fluido

Portalin näkymät ovat tästä syystä luotu hyväksikäyttämällä Bootstrappia ja sen valmiita komponentteja ja tyylejä.

Bootstrap on mahdollista asentaa Ruby on Rails -projektiin suoraan Geminä tai manuaalisesti lisäämällä tarvittavat tiedostot ja kansiot. Geminä asentamisen etuna on siistimpi projektin kansiorakenne, kun Bootstrapin ylimääräiset tiedostot ja kansiot eivät ole suoraan kehitysympäristön yleisimmissä kansioissa näkyvillä, eikä sen käyttöönotto vaadi erillistä tiedostojen kopioimista. Myös käytettävän Bootstrap-version hallinta muuttuu automaattiseksi, ellei sen sisältämän Gemin haluttua versiota määritetä erikseen. Bootstrap sisältää omat fontit, CSS-tyylitiedostot sekä JavaScript-kirjaston.

### 3.7 Heroku-pilvipalvelualusta

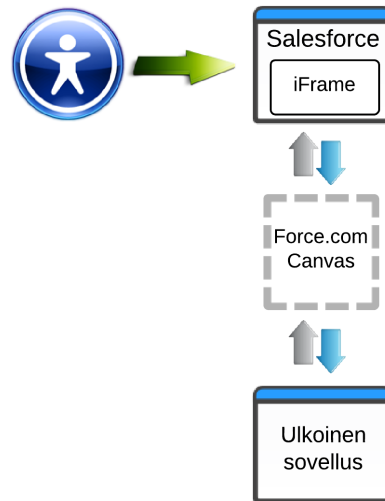
PaaS-tyyppinen pilvipalvelualusta Heroku tarjoaa ohjelmistokehittäjille ympäristön, jossa voidaan suorittaa useilla eri ohjelmointikielillä kehitettyjä sovelluksia. Tuettujen ohjelmointikielten joukosta löytyvät nykyään muun muassa Ruby, Java, Node.js, Scala, Clojure, Python, PHP ja Perl. Tämän ansiosta useammat kehittäjät voivat julkaista osaamilaan kielillä toteutettuja sovelluksia. Kehitetyt ohjelmat siirretään Herokuun hyödyntäen Git-versionhallintaohjelmistoa, jonka jälkeen ohjelmien asetuksia, ympäristömuuttujia ja valmiita lisäosia voidaan tarvittaessa muokata joko Herokun tarjoaman Toolbelt-työkalun avulla, tai käyttämällä verkkosivuston hallintaominaisuuksia. Heroku on yksi ensimmäisistä pilvipalvelualustoista, jota on kehitetty vuodesta 2007 lähtien. Vuonna 2010 se siirtyi Salesforce.comin omistukseen ja toimii nykyään sen alaisena tytäryhtiönä. [7]

Herokuun vietyjä sovelluksia voidaan suorittaa joko käyttäen alustan tarjoamia ilmaisia ominaisuuksia, tai tarvittaessa niiden kuukausimaksullisia versioita. Erilaisia ilmaisia ominaisuuksia on tarjolla suuri määrä, kuten käytettäviä tietokantajärjestelmiä ja niiden varmuuskopiointityökaluja. Kuukausiveloitukselliset ominaisuudet tarjoavat laajennettuja toiminnallisuuksia, kuten suurempaa tallennustilaa tietokantaan. Herokun *dyno* puolestaan tarjoavat enemmän suorituskykyä suoritettaville sovelluksille. Fluido Portalin kehitysversiossa ei ole käytössä maksullisia ominaisuuksia, mutta sen julkaisuversiossa täytyy harkita niiden tarpeellisuutta.



### 3.8 Force.com Canvas -ohjelmistokehys

Salesforce.com julkaisi Force.com Canvas -ohjelmistokehysten Winter '13 -julkaisunsa mukana. Se tarjoaa paketin työkaluja ja JavaScript-ohjelmointirajapintoja, joiden avulla voidaan kehittää kolmannen osapuolen ohjelmia ja näyttää ne Salesforce.comissa. Havainnollistus Force.com Canvas -sovelluksen toiminnasta ja sen käytöstä esitetään kuvassa 2.



Kuva 2: Force.com Canvas -sovellus Salesforce.comissa.

Työkaluilla on mahdollista luoda vuorovaikutteisuus käytettävän Salesforce.com-instanssin ja Canvas-sovelluksen välille, jonka avulla sovellus voi hakea tietoa Salesforce.com-instanssista käyttöönsä ja myös luoda sinne esimerkiksi uusia *Facebook*-päivityksiin verrattavissa olevia *Chatter*-päivityksiä. Force.com Canvas ei rajoita käytettävää ohjelmointikieltä, mutta vaatii, että sen käytössä on HTTPS-yhteys [8].

Tarve Force.com Canvasin käyttöön löytyy usein erillisten ulkoisten järjestelmien käytön luomasta ongelmasta, jossa käyttäjä joutuu hallitsemaan useissakin eri paikoissa samaan aikaan yhteen kokonaisuuteen liittyviä toimintoja. Esimerkkinä voi olla jokin verkkopalvelu, jonka sisältämä data tuotetaan toisessa järjestelmässä, kuten Salesforce.comissa. Ulkoista järjestelmää käyttämällä voidaan kiertää Salesforce.comin asettamia haasteita, kuten tallennustilan määrään liittyvät rajoitukset. Koska Salesforce.comin tarjoama tallennustila on erittäin kallista ja rajoitettua, ei *Big Data* -tyyliset ratkaisut ole alustan sisällä vielä mahdollisia. Usein kyseessä voi myös olla jokin yrityksen tärkeä järjestelmä, joka on kehitetty ennen Salesforce.com-palvelun käyttöönottoa, eikä sen tarjoamia vastaavia toiminnallisuuksia ole erikseen kehitetty Salesforce.comin sisäisesti.

Tietoturvan puolesta on myös mahdollistaa mukauttaa Force.com Canvasia käyttävää kolmannen osapuolen sovellusta niin, että Salesforce.com-palvelun käyttäjä voi automaattisesti käyttää sitä Salesforce.comin sisällä ilman erillistä kirjautumista järjestelmään. Force.com Canvas sovelluksien käyttöoikeuksia voidaan hallinnoida Salesforce.comin sisäisesti, jonka lisäksi sovellus voi hakea erillistä tietoa Force.com Canvasin kautta myös muun muassa käyttäjistä. Tämä siis myös helpottaa kolmannen osapuolen sovelluksen ylläpidettävyyttä, kun tarvittavia oikeuksia voidaan hallinnoida yhdessä paikassa. Force.com Canvasin käyttöönotto ei vaadi välttämättä suuria muutoksia sitä hyödyntävään sovellukseen, mutta sen tarjoamien toiminnallisuuden tutkiminen ennen toteuttamista on tärkeää, jotta toiminnallisuuden ja käytettävyyden kannalta paras ratkaisu voidaan saavuttaa.

### 3.9 GIT-versionhallinta

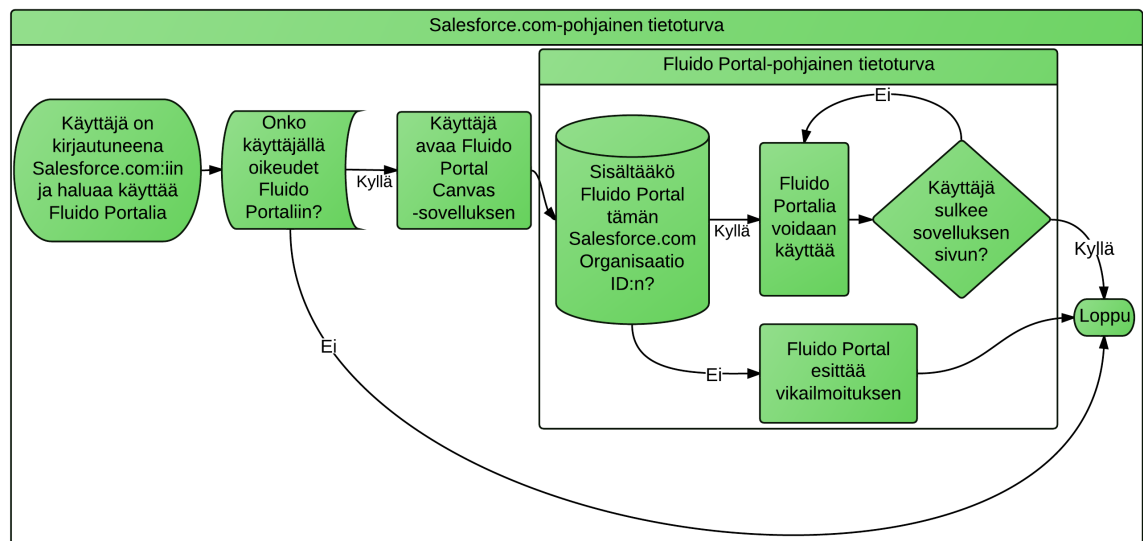
Linus Torvalds alkoi vuonna 2005 suunnittelemaan ja kehittämään uutta versionhallintajärjestelmää Linuxin ytimen kehitystyötä varten. Tästä sai alkunsa projekti, jonka nimeksi annettiin *Git*. Gitissä on panostettu nopeuteen ja sen toimintaan hajautetussa kehitystyössä, jossa useat eri kehittäjät voivat päivittää tiheään tahtiin samaa työstettävää projektia. Vaatimuksina Gitille oli myös datan virheellisuuden ja sen katoamisen estäminen, sekä mahdollisuus hallita valtavia projekteja kuten Linux.

Gitin ominaisuuksiin kuuluu myös useiden eri kehityshaarojen tehokas ja helppo hyödyntäminen, joissa eri kehittäjät voivat säilyttää omia kehitysversioitaan. Työstettyään toiminnallisuuden he voivat yhdistää haarojen eroavaisuudet ja päivittää lopuksi muutokset päähaaraan. Rakenteeltaan Git on käytännössä joukko ohjelmia, jotka tekevät eri toiminnallisuuden [9]. Se on noussut erittäin suosituksi versionhallintajärjestelmäksi, jota käytetään nykyään suurimmassa osassa uusia projekteja. Git on myös pakollinen työkalu kehitettäessä Heroku-sovelluksia.

## 4 Autentikointi ja tietoturva

### 4.1 Autentikoinnin ja tietoturvan yleiskuvaus Fluido Portalissa

Asiakkaiden ja Fluido Oy:n välisten salaisten tietojen turvallinen säilyttäminen on erittäin tärkeää, sillä ulkoiset tahot voisivat väärinkäyttää tietoturva-aukkojen kautta saamia tietoja haitallisesti. Väärinkäytön riski digitaalisissa sovelluksissa täytyy siis minimoida käyttämällä turvallisiksi todettuja tekniikoita, kuten yleisesti käytettäviä autentikointimenetelmiä ja rajoituksia. Fluido Portalin autentikointitapa poikkeaa perinteisestä käyttäjätunnus/salasana-yhdistelmän menetelmästä merkittävästi, sillä sen tietoturva on suunniteltu hyödyntämään Salesforce.com-alustan ja Force.com Canvas -rajapinnan tarjoamia tietoturvaratkaisuja yhdessä sovelluksen sisäisten estojen kanssa.



Kuva 3: Tietoturvasojen kuvaus käytön aikaisesti.

Yllä esitetystä kuvaajasta (kuva 3) pyritään esittämään teknisestä- ja käyttäjän näkökulmasta, miten eri tietoturvan tasot vaikuttavat Fluido Portalin käyttöön. Tietoturvan tasoiksi määritellään tässä tapauksessa Salesforce.comin omat tietoturva-asetukset, sekä sovelluksen sisäänrakennetut tietoturva-asetukset. Koska asiakkaat käyttävät omia Salesforce.com-instanssejaan, jotka vaativat tunnistautumisen luotettavaksi luokiteltavaan järjestelmään, voidaan heihin näin ollen luottaa. Tämän pohjalta Fluido Portalin autentikoinnin riittävä turvallisuustaso saavutetaan hyväksikäyttämällä Force.com Canvas

-rajapinnan tarjoamia tietoja, jotka portaali saa vastaansa asiakkaan sen avatessa. Tarvitavana tietona riittää Salesforce.com-instanssin uniikki *Id*, jonka tulee löytyä sovelluksen tietokannasta. Kyseinen *Id* pyydetään erikseen asiakkaalta, jolle Fluido Portal asennetaan käytettäväksi, ja lisätään Fluido Oy:n toimesta tietokantaan. Kun *Id* löytyy tietokannasta, pääsee asiakas käyttämään sovellusta. Tunnistauduttuaan asiakas voi nähdä vain omaan organisaatioonsa liittyvää dataa. Sovelluksen sisäinen logiikka suodattaa organisaation *Id*:n perusteella tietokannasta sallittua dataa käyttöön, jonka perusteella myös mahdolliset lisäksi tarvittavat tiedot voidaan hakea. Edellinen kuvaaja ei kuitenkaan paljasta kaikkia autentikointiin liittyviä tapahtumia, joita pellin alla tapahtuu, kuten Force.com Canvasin sisäistä autentikaatiota.

#### 4.2 Force.com Canvas -rajapinnan autentikaatio

Fluido Portalin toimimisen edellytyksenä on sen hyödyntämän Force.com Canvasin oman sisäisen autentikoinnin toteuttaminen. Salesforce.comin ja kolmannen osapuolen sovelluksen, tässä tapauksessa Fluido Portal, välillä toimiva Force.com Canvas vaatii *Signed Request* -, tai *OAuth2*-tyyppisen autentikoinnin, jonka avulla se päättää, voidaanko sovellusta käyttää. Oletusarvoisesti Force.com Canvasin käytössä suositellaan *Signed Request* -tyyppisen autentikointimenetelmän käyttämistä, jonka palauttamia tietoja voidaan hyödyntää kaikissa jatkossa esiintyvissä sovelluksen lähettämissä pyynnöissä Salesforce.comin suuntaan. [8]

Käytettäessä *Signed Request* -tyyppistä autentikointimenetelmää, Force.com Canvas -sovellusta käynnistettäessä Salesforce.comin puolella, se lähettää automaattisesti *Signed Request* -pyynnön kolmannen osapuolen sovellukselle. Sovellus keskustelee Salesforce.com-puolen kanssa ja päättää, voidaanko sovellusta käyttää. Pyyntö lähetetään Salesforce.comin puolelta HTTP POST -metodilla, joka sisältää *string*-tyyppisenä tarvittavat tiedot, kuten *access tokenin*. Kolmannen osapuolen sovelluksen täytyy toteuttaa tarvittava toiminnallisuus pyynnön käsittelyä varten, jonka Force.com Canvas -rajapinta tarjoaa sen käyttöön. Jos käytössä olisi *OAuth2*-tyyppinen autentikointi, käyttäjä näkisi ponnahdusikkunan, jossa pyydetään lupa sovelluksen autentikointiin [8]. Tämän vuoksi on yksinkertaisempaa ja selvempää käyttää *Signed Request* -tyyppistä autentikointitapaa, jolloin loppukäyttäjältä ei vaadita ylimääräistä toimintaa sovelluksen käyttämiseksi.

Fluido Portalissa Force.com Canvasin vaatimaa autentikointia varten on luotu erillinen yleispätevä Gem-laajennus toisen sovelluksen kehittäjän toimesta, joka toteuttaa autentikointiin liittyvät vaatimukset, tarjoten myös samalla joitain apufunktioita sen käyttöä varten. Gem-laajennuksen käyttämä autentikointitapa hyödyntää *Signed Request* -tyyppistä autentikaatiota, jonka palauttamilla tiedoilla vuorovaikutteisuutta Fluido Portalin ja Salesforce.comin välillä on mahdollista käyttää. Vuorovaikutteisuudella tarkoitetaan tässä tapauksessa tietojen hakemista Salesforce.com-puolelta, sekä mahdollisesti uusien tietueiden luomista Salesforce.com-puolelle. Tehtäessä vuorovaikutteisia pyyntöjä Force.com Canvas -rajapinnan tarjoamien JavaScript-kirjastojen avulla, käytetään niissä autentikoinnissa saatua *Signed Request*-objektia. Kyseinen Gem-laajennus on Fluido Oy:n sisäisessä käytössä, eikä sitä ole julkaistu yleiseen käyttöön. Sitä voidaan jatkossa hyödyntää myös mahdollisten muiden Force.com Canvasia hyödyntävien Ruby on Rails -sovellusten kehittämisessä.

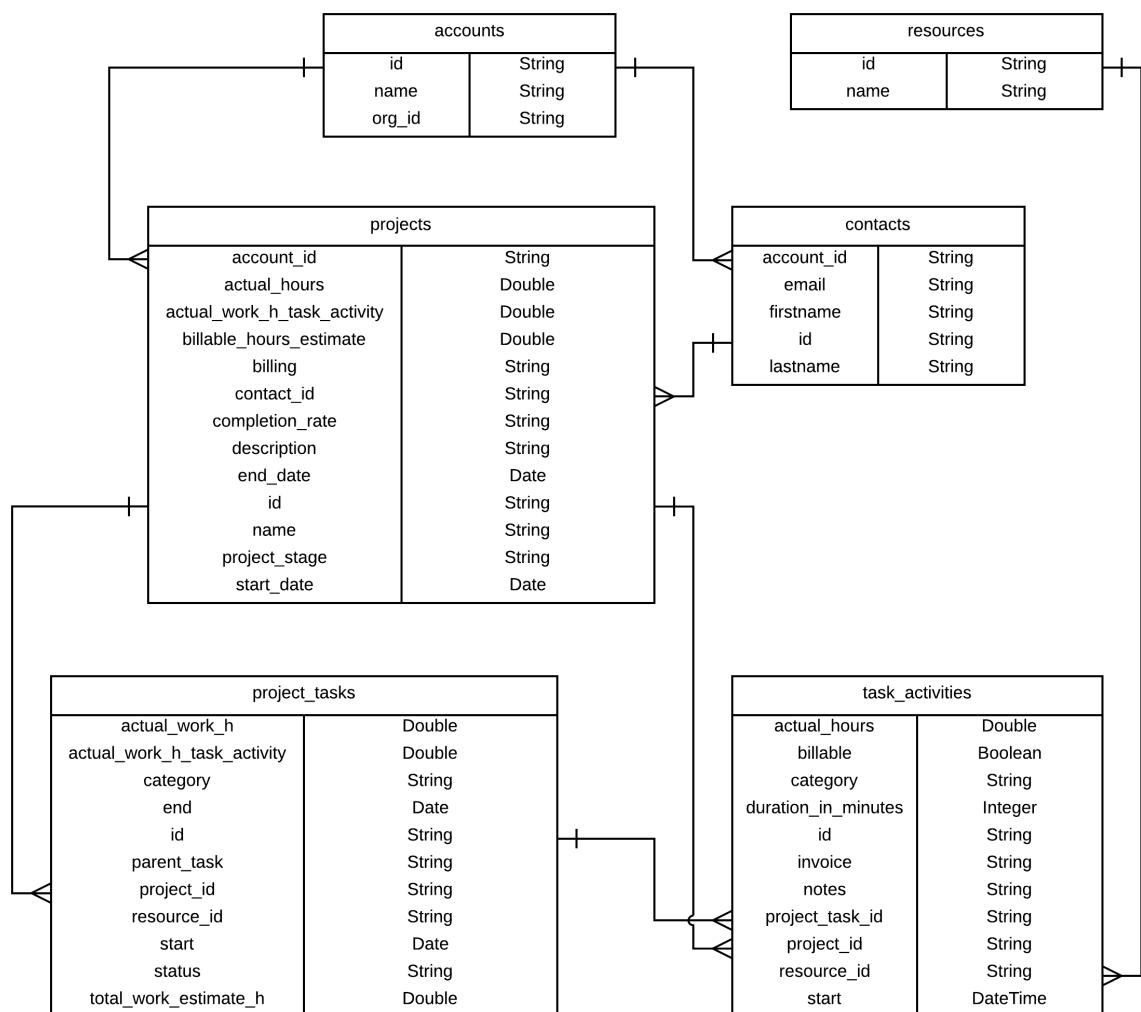
#### 4.3 Salesforce.com-puolen käyttöoikeuksien hallinta

Kun asiakas on asentanut Fluido Portalin asennuspaketin omaan Salesforce.com-instanssiinsa, sovellus on heti käyttövalmis. Asennuksen aikana, tai sen jälkeen, asiakas pystyy itse määrittämään, ketkä käyttäjistä voivat sitä käyttää. Salesforce.com-ympäristön sisäinen käyttöoikeuksien hallinta mahdollistaa monipuolisen säätövaran erilaisiin oikeuksiin, kuten tässä tapauksessa sovelluksen käyttöön. Käyttöoikeuksien kriteereinä voidaan soveltaa muun muassa käyttäjän profiliin ja rooliin liittyviä turvallisuusasetuksia, tai henkilökohtaisesti jaettuja oikeuksia. Koska näihin asetuksiin voidaan vaikuttaa vain siinä Salesforce.com-instanssissa, mihin Fluido Portalin asennuspaketti on asennettu, siirtyy vastuu tietoturva-asetuksien määrittämisestä asiakkalle. Salesforce.com-instanssin ulkopuolelta kyseisiä asetuksia ei ole mahdollista muuttaa.

## 5 Fluido Portalin tietokantamalli

### 5.1 Tietorakenne

Kaiken tiedon perustana on jonkinlainen lähde, josta se muodostuu, jonka jälkeen sitä voidaan kysellä ja käyttää erilaisiin tarkoituksiin. Tietoteknisissä sovelluksissa tarvittavaa tietoa voidaan hakea erilaisin menetelmin, mutta yksi yleisimmistä lähteistä sille on jokin tietokanta, jossa siihen määrättyä dataa säilytetään. Fluido Portalin käyttämä PostgreSQL-tietokantajärjestelmä sisältää tämänhetkisessä kehitysvaiheessa suhteellisen yksinkertaisen tietokantamallin, missä asiakkaiden projekteihin liittyvää dataa säilytetään.



Kuva 4: ER-kaavio tietokannan rakenteesta.

Edellisellä sivulla esitetty kuva 4 kuvaa tietokannan rakennetta. Se sisältää kuusi taulua, jotka sisältävät viittauksia toisiinsa. Fluido Portalin toiminnan puolesta tärkein taulu on *accounts*, joka vastaa Salesforce.comin Account-objekteja. Asiakkaan kirjautuessa omasta Salesforce.com-instanssistaan Force.com Canvas välittää mukanaan *Organization Id*:n, jota verrataan *accounts*-taulun sisältämiin *org\_id*-kenttiin. Kytköksen löytyessä käyttäjälle pystytään hakemaan tietoja projekteista, jotka ovat kyseisen *org\_id*:n sisältävän tilin (account) omistamia. Tämän toiminnallisuuden mahdollistaa taulujen sisältämät viitekentät. Esimerkkinä viitteistä voidaan tutkia projekteja vastaavan *projects*-taulun kenttiä, joista yksi on *account\_id*. Tämä toimii kytköksenä yhteen *accounts*-taulun tietueeseen, joka vastaa asiakasta koskevaa tiliä. Samalla toimintaperiaatteella saadaan haettua myös projektien omistamia tietueita muista tauluista, kuten tuntimerkintöjä, joita tietokannassa vastaa taulu *task\_activities*. ER-kaaviota tutkimalla voidaan huomata, että kaikki muut taulut ovat *accounts*-taulun alla, paitsi *resources*, joka kuvaa työtä tehneitä toimihenkilöitä. Tämä johtuu siitä, että toimihenkilöt eivät yleensä ole kytköksissä vain johonkin tiettyyn tiliin. Tuntimerkinnöillä on viittaus toimihenkilöön, mutta toimihenkilöllä voi olla useampia tuntimerkintöjä useissa eri tilien projekteissa.

## 5.2 Tietokannan migraatietiedostot

Tietokantatauluissa käytettävät pääavaimet, eli *id*-kentät, muodostetaan PostgreSQL-tietokannoissa normaalisti automaattisesti tietueen luonnin yhteydessä Integer-tyyppisinä. Fluido Portalissa on kuitenkin oleellista, että Salesforce.comista haettujen tietueiden *Id*:t saataisiin asetettua pääavaimiksi. Jotta tämä onnistuu, vaatii se muutoksien toteuttamista Ruby on Rails -sovelluksen tietokantaa muokkaavia migraatietiedostoja käyttäen.

```

20140212113357_change_primary_keys.rb
1  class ChangePrimaryKeys < ActiveRecord::Migration
2    def up
3      remove_column :accounts, :id # remove existing primary key
4      remove_column :contacts, :id # remove existing primary key
5      remove_column :projects, :id # remove existing primary key
6      remove_column :project_tasks, :id # remove existing primary key
7      remove_column :resources, :id # remove existing primary key
8      remove_column :task_activities, :id # remove existing primary key
9      rename_column :accounts, :sf_id, :id # rename existing UDID column
10     rename_column :contacts, :sf_id, :id # rename existing UDID column
11     rename_column :projects, :sf_id, :id # rename existing UDID column
12     rename_column :project_tasks, :sf_id, :id # rename existing UDID column
13     rename_column :resources, :sf_id, :id # rename existing UDID column
14     rename_column :task_activities, :sf_id, :id # rename existing UDID column
15     execute "ALTER TABLE accounts ADD PRIMARY KEY (id);"
16     execute "ALTER TABLE contacts ADD PRIMARY KEY (id);"
17     execute "ALTER TABLE projects ADD PRIMARY KEY (id);"
18     execute "ALTER TABLE project_tasks ADD PRIMARY KEY (id);"
19     execute "ALTER TABLE resources ADD PRIMARY KEY (id);"
20     execute "ALTER TABLE task_activities ADD PRIMARY KEY (id);"
21   end
22
23   def down
24     execute "ALTER TABLE accounts DROP CONSTRAINT table_pkey;"
25     execute "ALTER TABLE contacts DROP CONSTRAINT table_pkey;"
26     execute "ALTER TABLE projects DROP CONSTRAINT table_pkey;"
27     execute "ALTER TABLE project_tasks DROP CONSTRAINT table_pkey;"
28     execute "ALTER TABLE resources DROP CONSTRAINT table_pkey;"
29     execute "ALTER TABLE task_activities DROP CONSTRAINT table_pkey;"
30     rename_column :accounts, :id, :sf_id
31     rename_column :contacts, :id, :sf_id
32     rename_column :projects, :id, :sf_id
33     rename_column :project_tasks, :id, :sf_id
34     rename_column :resources, :id, :sf_id
35     rename_column :task_activities, :id, :sf_id
36     add_column :accounts, :id, :primary_key
37     add_column :contacts, :id, :primary_key
38     add_column :projects, :id, :primary_key
39     add_column :project_tasks, :id, :primary_key
40     add_column :resources, :id, :primary_key
41     add_column :task_activities, :id, :primary_key
42   end
43 end

```

Line 43, Column 4      Spaces: 4      Ruby

Kuva 5: Esimerkki Rubyn migraatiotiedostosta.

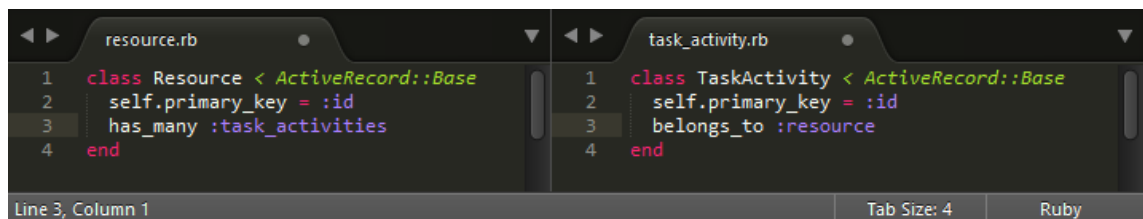
Edellä mainitun ongelman ratkaisemiseksi on luotu yllä esitetyn kuvan 5 mukainen migraatiotiedosto, missä taulujen alkuperäiset *Id*-kentät korvataan aiemmin *sf\_id*-nimisiksi luoduilla kentillä. Tämän avulla myös kenttien alkuperäinen tyyppi *Integer* on nyt saatu muutettua *String*-tyyppiin, joka näin ollen tukee Salesforce.comin kaltaisia *Id*:itä. Tutkimalla migraatiotiedostoa voidaan huomata osiot *def up* ja *def down*. Näissä osioissa on mahdollisuus määrittää sen tekemät muutokset tietokantaan, kun kyseessä on joko uusien määritysten luominen (*def up*-osio) tai jo ennestään saman migraatiotiedoston tekemien muutosten palauttaminen aiempaan tilaan (*def down*-osio).



Migraatietiedostojen käyttö on pakollista myös silloin, kun tietokantaan halutaan lisätä jokin uusi objektia vastaava taulu tai sen kenttä. Jos Fluido Portaliin haluttaisiin jatkossa lisätä esimerkiksi Salesforce.comin mahdollisuuksia (*opportunity*) vastaavia objekteja, luotaisiin sitä varten uusi migraatietiedosto. Rubyn nimeämislogiikan kannalta on huomiotava, että uutta tietokannan taulua määritettäessä, objektin nimi tulee antaa monikkomuodossa. Esimerkiksi objektiin *opportunity* viitattaisiin migraatietiedostossa nimellä *opportunities*. Joskus monikkumuotojen käyttäminen saattaa aiheuttaa ongelmia, kun sanan alkuperäiskieli ei ole englanti, tai kyseessä on jokin erikoisempi sana, jota Ruby ei osaa kääntää oikeaan monikkumuotoon.

### 5.3 Assosiaatioiden luominen ja Active Record

Ruby on Rails -ohjelmissa voidaan hyödyntää *Active Record* -arkkitehtuurimallia, jonka avulla toisiinsa liittyvien objektien väliset riippuvuudet voidaan määrittää. Riippuvuudet voidaan luoda määrittämällä niiden väliset suhteet, eli assosiaatiot, joiden avulla toisiinsa kytkeytyneiden objektien väliset viittaukset voidaan toteuttaa helposti.



Kuva 6: Assosiaatioiden luonti ruby-luokissa.

Kuten koodikatkelmasta voidaan huomata (kuva 6), Active Recordia käytettäessä kuvataan tietokantatauluja vastaavien objektien relaatiot niiden omissa Ruby-luokissa, eli malleissa (model). Relaatiot kuvataan määrittämällä objektien väliset suhteet esimerkiksi määritteillä *has\_one* ja *has\_many* (omistaa yhden tai useamman) sekä *belongs\_to* (kuuluu johonkin), jonka jälkeen määritellään, mikä objekti on kyseessä avainsanan kanssa. Portaaliprojektin objekteista tuntimerkinnät (*task\_activities*) ja toimihenkilöt (*resources*) ovat kytköksissä toisiinsa suhteilla, jossa toimihenkilöllä voi olla useita tuntimerkintöjä, mutta tuntimerkintä kuuluu vain yhdelle toimihenkilölle. Kuten migraatietiedostojen osallakin, täytyy objektien yksikkö- ja monikkumuotoihin kiinnittää huomiota, sillä Rubyssa assosiaatioita luotaessa niiden suhteen laatu määrittää käytettävän muodon.

Kytköksiä luomisen jälkeen tietueita haettaessa voidaan automaattisesti hakea myös niihin kytketyt objektit. Tämä auttaa huomattavasti etenkin tilanteessa, jossa verkkosivulla yritetään esittää tietoja, jotka vaativat dataa useamman eri objektin kentistä. Fluido Portalissa tätä ominaisuutta hyödynnetään tilanteessa, jossa tuntimerkintä objektin omien kenttätietojen lisäksi halutaan esittää myös sen tekijän nimitiedot. Viittaus nimeen onnistuu nyt helposti käyttämällä pistenotaatiota, kuten `'tuntimerkintä.resurssi.nimi'`. Active Recordin käyttäminen on suoraviivaista, ja yksinkertaistaa kirjoitettavaa sovelluslogiikkaa. Ilman Active Recordia kehittäjä joutuisi toteuttamaan erillisiä kyselyitä eri objekteja vastaaviin tauluihin, ja vertaamaan niiden yhteyksiä toisiinsa käyttäen *map*-toiminnallisuuksia vastaavia menetelmiä.

#### 5.4 Datan synkronointi Salesforce.com-instanssin kanssa

Asiakkaiden projekteihin liittyvät tiedot päivittyvät aina ensiksi Fluidon oman Salesforce-instanssin tietokantaan, jota ei käyttörajoitusten ja viiveiden vuoksi ole suoraan kytketty Fluido Portalin käyttöön. Jotta Fluido Portalin oman tietokannan tiedot saadaan vastaamaan Fluidon Salesforce.com-instanssin tietokantaa, tarvitaan siihen jonkinlainen synkronointiohjelmisto väliin. Tätä tarvetta varten on rakennettu mahdollisimman yksinkertainen väliaikaisratkaisu Ruby on Rails -sovelluksen muodossa, jolla tietokantaa voidaan päivittää yksisuuntaisesti. Ohjelma pyörii Fluido Portalin tavoin Herokun päällä, mutta on eriytettyä pääsovelluksesta. Tästä huolimatta se jakaa saman tietokannan, jonka vuoksi dataa on helppo päivittää. Eriytys Fluido Portalista on tehty tietoisena ratkaisuna, jonka johdosta toiminnallisuuden mahdollinen vaihto parempaan sujuu yksinkertaisemmin ja siistimmin, kun vanhoja koodia ei tarvitse siivota pois pääprojektista.

Synkronoinnin automatisointia varten on sovellukseen kehitetty *Rake*-toiminnallisuus, joka voidaan ajastaa pyörimään halutuin väliajoin Herokun tarjoamia valmiita toiminnallisuksia hyväksikäyttäen. *Rake*-toiminnallisuus toteutetaan luomalla sitä varten toiminnallista logiikkaa sisältävä Ruby-luokka, jota suoritetaan Herokun ajastinta (*scheduler*) hyväksikäyttäen. Automaattinen synkronointi poistaa manuaalisen päivittämisen tarpeen, samalla varmistaen tuoreen datan viemisen Fluido Portalin käytettäväksi. Tällä hetkellä ajastin on asetettu suorittamaan päivitysoperaatio kerran päivässä, joka tarjoaa projekti-tietoja varten riittävän päivitystiheyden, sillä niitä ei ole tarkoitus jakaa reaaliaikaisesti.

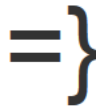
Samalla voidaan varmistaa, että Salesforce.comin asettamat käyttörajoitukset tietokantaoperaatioille eivät tule vastaan, kun Herokuun tuotavien päivitysten tiheys pidetään hillittynä.

Automaattisen päivitystoiminnallisuuden lisäksi etenkin kehitysvaiheessa on kuitenkin myös tarpeellista viedä tuoreeltaan päivitettyjä tietoja suoraan tietokantaan. Tämän vuoksi synkronointisovellus tukee *Rake*-toiminnallisuuden suorittaman kaikkien taulujen päivityksen lisäksi myös yksittäisten taulujen päivittämistä sovelluksen verkkosivusto-pohjaisen käyttöliittymän avulla. *Rake*-toiminnallisuuden logiikka muodostuu käytännössä näiden käyttöliittymässä toteutettavien yksittäisten päivitysten toteuttavan logiikan yhdistämisestä yhteen Ruby-luokkaan.

**fluido.**

## Portal Ultimate Database Synchronizer TURBO 4000

Synchronize Accounts
Synchronize Projects
Synchronize Resources
Synchronize Task Activities



Copyright © 2014 FLUIDO LTD. | Salesforce CRM-Ratkaisut ja Konsultointi | Tekniikantie 14, 02150 Espoo, Finland | +358 6 781 2121

Kuva 7: Fluido Portal Synchronizer -sovelluksen käyttöliittymä.

Yllä esitetyn kuvan 7 käyttöliittymän painikkeet laukaisevat tarvittavat toiminnallisuudet taulujen yksittäistä päivittämistä varten. Toteutuksessa jokainen painike kutsuu uutta näkymää, johon on kytketty omat toiminnalliset logiikkansa sisältävät funktiot. Kun päivitys on valmis, näkymä avautuu käyttäjälle sisältäen ilmoituksen toiminnon onnistumisesta. Jos päivitys epäonnistuu, tulee näkyviin Ruby on Railsin oma virhesivu, jossa esitetään ongelman tekninen kuvaus.

Toisin kuin Fluido Portal -pääohjelmassa, Force.com Canvas -rajapintaa ei hyödynnetä synkronointisovelluksen käytössä tässä kehitysvaiheessa. Sen sijaan synkronointiohjelmassa käytetään hyväksi *basedotcom*-Gemiä, jonka avulla ohjelma pystyy autentikoitumaan OAuth2-protokollaa käyttäen Salesforce.com-instanssiin. Onnistuneen autentikoitumisen jälkeen dataa pystytään hakemaan Gemin toiminnallisuuksia käyttäen automaattisesti REST-rajapinnan avulla. Jotta Gem hakee halutut tiedot Salesforce.com-instanssista, vaatii se yhden funktionsa parametrinä *SOQL*-kyselylauseen, joka on Salesforce.comissa

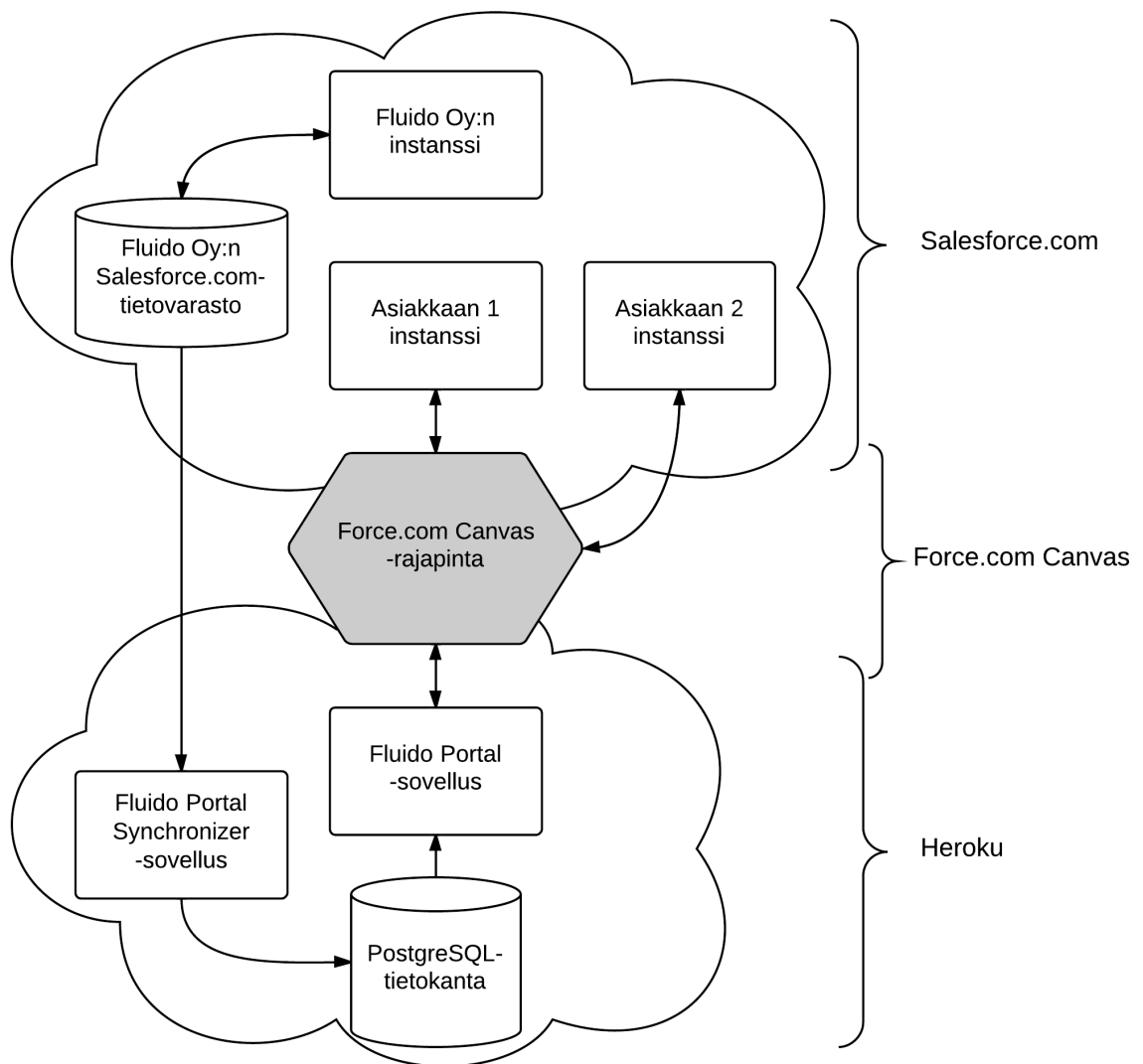
käytettävän *APEX*-ohjelmointikielen käyttämä kieli tietokannan hakukyselyissä. Funktion suorittamisen jälkeen Gem *'materialisoi'* haetut datat Rails-sovelluksen paikallisiin muuttujiin. Kun valittu data on käytettävissä, se iteroidaan ohjelmallisesti läpi ja parsitaan SQL-kyselylausemuotoon tietokantamallin taulujen mukaisesti.

Kyselylauseilla muodostetaan joukko *INSERT*-operaatioita, jotka syötetään tietokantaan, kunhan tietokannan taulujen vanhat datat on tyhjennetty. Syy tyhjennykseen löytyy toteutustavan logiikasta, sillä mahdollisesti poistuneen ja päivitetyn datan synkronointi vaatisi UPSERT-kyselylauseiden toteuttamista sekä vertailuoperaatioita Herokun PostgreSQL-tietokannan ja Salesforce.com-instanssin datavarantojen välillä. Toteutustapa on osittain alkeellinen, mutta liiallinen työajan käyttö sen toteuttamiseen olisi ollut turhaa. Tämä siksi, että tulevaisuudessa on ilmestymässä Salesforce.comin tarjoma virallinen, tämän toiminnallisuuden korvaava välikappale (*CloudConnect*) Herokun ja Salesforce.comin välille, joka synkronoi tietokantojen dataa huomattavasti edistyneemmin keinoin.

## 6 Fluido Portalin kehitysversio ja jatkokehityssuunnitelmat

### 6.1 Sovelluksen arkkitehtuurikuvaus

Sovelluksen yleistä rakennetta ja toimintaa sen suoritussympäristössä voidaan havainnollistaa sopivalla abstraktiotasolla esitetyillä arkkitehtuurikuvauksilla, joiden tarjoma yleiskuva helpottaa kokonaisuuden hahmottamista.



Kuva 8: Fluido Portal -sovelluksen arkkitehtuurikuvaus

Fluido Portalin arkkitehtuurikuvaus (kuva 8) esittää sovelluksen yleisiä kytköksiä ja toimintaperiaatteita eri ympäristöjen välillä, sekä toiminnallisessa kokonaisuudessa käytettyjä ratkaisuja. Pääsovellus sijaitsee Herokun pilvipalvelualustan päällä yhdessä *Fluido*

*Portal Synchronizer* -sovelluksen kanssa, jonka toiminnallinen tarkoitus on synkronoida sovellusten yhteisessä käytössä olevan PostgreSQL-tietokannan tiedot vastaamaan tarvittavien tietojen osalta Salesforce.comissa sijaitsevan Fluidon tietokannan sisältämää dataa. Fluido Portal käyttää synkronointisovelluksen hakemaa dataa visualisoidessaan asiakkaiden projektien tietoja sovelluksen käyttäjille, mutta ei itse tee sinne tässä vaiheessa muutoksia. Synkronointisovellus puolestaan ei hae käyttöönsä tietokannan sisältämää dataa, vaan tekee sinne ainoastaan päivitysten ohella muutoksia.

Fluido Portalin käyttö tapahtuu Salesforce.com-ympäristöissä, jonka mahdollistamiseksi käytössä on Salesforce.comin tarjoama Force.com Canvas -rajapinta. Fluido Portalin asennuspaketti sisältää tarvittavat Force.com Canvas -sovelluksen vaatimat osat, kuten sen näkymän tarjoavan *Visualforce*-sivun ja osapuolten välisen yhteyden mahdollistavan Salesforce.comin *yhdistetyn sovelluksen* (*Connected App*). Rajapintaa käyttävän sovelluksen ja Salesforce.com-instanssin välisen rajapinnan sisäisen autentikoinnin jälkeen sovelluksen tarjoama instanssi on käytettävissä, jonka välittäjänä Force.com Canvas toimii. Sovellus voidaan näyttää *Visualforce*-sivulla sitä varten luodun *iFrame*-tekniikkaa hyödyntävän tilan sisällä, jonka lisäksi se pystyy keskustelemaan sitä käyttävän Salesforce.com-instanssin kanssa Force.com Canvas -rajapinnan tarjoamien toimintojen avulla. Keskustelukanava on kaksisuuntainen, joka mahdollistaa erilaisten rajapinnan tarjoamien toiminnallisuuksien käytön puolin ja toisin. Näin ollen esimerkiksi Fluido Portal voi kysellä itsensä sisältävän *iFramen* leveys- ja korkeustietoja käyttäjän näytöllä, ja pyytää myös tarvittaessa niiden arvojen muuttamiseen vaikuttavaa funktiota Salesforce.comin puolella.

Arkkitehtuurikuvauksesta voidaan myös huomata, ettei Fluido Portal ole tämän esimerkin osalta kytkettynä Fluido Oy:n omaan Salesforce.com-instanssiin. Syy kytköksen puuttumiseen löytyy sovelluksessa käytettävän logiikan ja käyttötarkoituksien puolelta. Kaikki projekteihin liittyvät tuoreimmat tiedot kokonaisuudessaan ovat jo ennestään tutkittavissa Salesforce.comin oman käyttöliittymän puolella, jolloin ulkoisen sovelluksen käyttäminen tähän tarkoitukseen on epäkäytännöllistä. Sovelluksen logiikka asettaa myös omat ongelmansa, sillä oletusarvoisesti Fluido Portalissa näytettävät tiedot ovat sidottuina asiakaskohtaisesti, eikä Fluidolle ole erikseen asetettu yleispätevää tunnusta kaikkien asiakkaiden projektien selaamiseksi. Myöskään tarvittavia suodattimia sovelluksen tarjoamien toimintojen järkevää käyttöä varten ei ole vielä toteutettu.

## 6.2 Sovelluksen tiedosto- ja kansiorakenne

Ruby on Rails -pohjaisilla sovelluksilla, kuten Fluido Portal, on tietynlainen perusrakenne siinä käytettyjen kansioden ja tiedostojen suhteen, jota kehitettävien sovellusten tulee noudattaa. Eri kansiot ovat kategorisoitu ja nimetty tietyn tarkoituksensa mukaisesti, jotta sovelluskehittäjille olisi selvempää mistä tarvittavat tiedostot löytyvät. Riippuen kehitettävästä sovelluksesta ei jokaisen kansion sisältämiin tiedostoihin ole tarvetta tehdä muutoksia, tai edes selata niiden sisältöä.



Kuva 9: Fluido Portal -sovelluksen tiedosto-/kansiorakenteen esittely.

Fluido Portal -sovelluksen kansiorakenne (kuva 9) vastaa Ruby on Rails -sovellusten yleistä kansiorakennetta täysin, eikä sinne ole lisätty MVC-arkkitehtuurin ulkopuolisia lisäkansioita. Kuvassa on korostettu Fluido Portalin kehitystyön osalta yleisimmin käytetyt kansiot sekä esitelty niiden sisältö. MVC-suunnitelumallia hyödyntävä Ruby on Rails toteuttaa mainittua mallia käyttäen erillisten kansioden *models*, *views* ja *controllers* sisältöjä, jotka sijaitsevat sovelluksen *app*-kansiossa. Fluido Portalin kohdalla mallit sisältävä *models*-kansio pitää sisällään objektien, jotka tässä tapauksessa vastaavat tietokannan sisältämiä tauluja, sisäiset määrittelyt, kuten assosiaatiot toisiin objekteihin. Ohjain-kansiossa (*controllers*), on näkymien (*views*) ja mallien välisen logiikan toteuttavat ohjain-luokat,

joilla esimerkiksi käyttöliittymän sivua ladattaessa voidaan suorittaa tarvittavien objektien hakukyselyitä. Näkymät tarjoava *views*-kansio puolestaan sisältää kaikki käyttöliittymän ulkoasun muodostavat HTML-pohjaiset tiedostot. Sovellusta siis pääosin kehitetään näissä kansioissa, sillä ne vastaavat toiminnallisen logiikan ja käyttöliittymän näkymien säilyttämisestä.

Kansio *config* puolestaan sisältää sovelluksen yleisiin asetuksiin liittyviä tiedostoja, joiden muokkaamisilla on usein merkittäviäkin vaikutuksia sovelluksen toimimisen kannalta. Yksi yleisimmin muokattavista tiedostoista *config*-kansion sisällä on *routes.rb*-luokka, jota ilman käyttöliittymä ei voi toimia. Se sisältää muun muassa määrittelyn käyttöliittymän aloitusnäköymästä sekä kutsuttavien linkkien tarvitsemat reititykset. Ruby on Railsissa reititykset tarkoittavat jonkin toiminnallisuuden toteuttamista, joka lopulta päättyy halutun näkymän näyttämiseen [10]. Myös sovelluksen käyttämien Gem-laajennusten määrittelytiedostot löytyvät usein *config*-kansioista, joihin voivat kuulua muun muassa kehitys- ja tuotantoympäristöissä käytettävien tietokantojen määrittelyt.

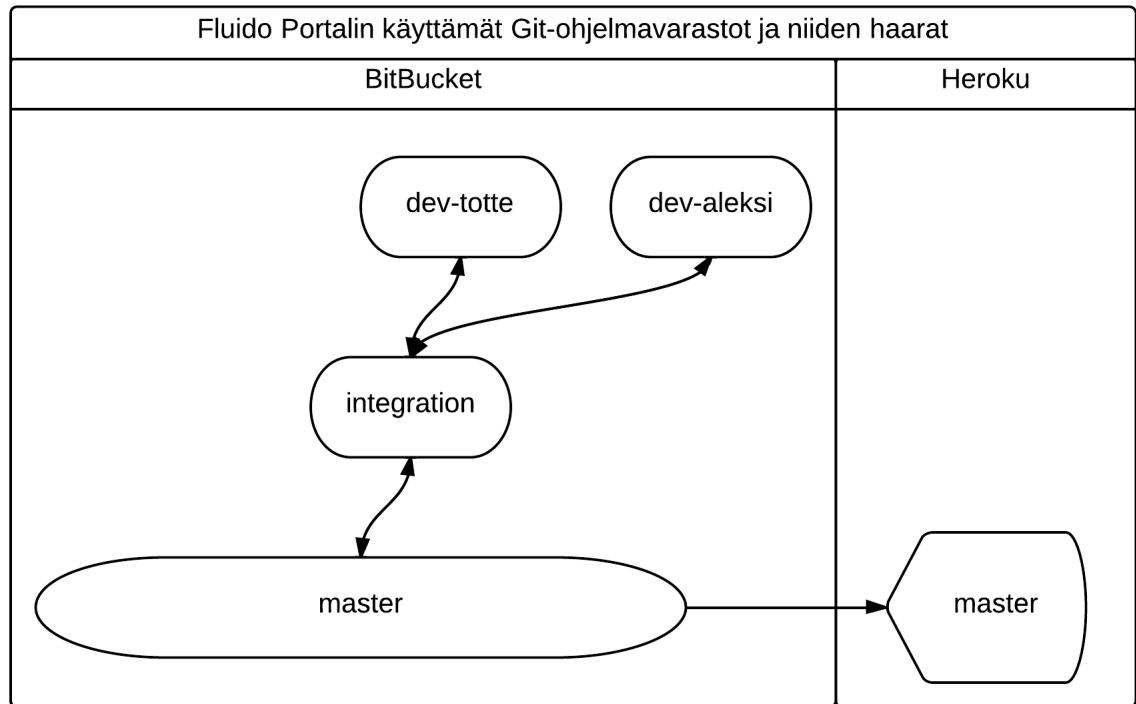
Tietokantaan liittyviä tiedostoja sisältävä *db*-kansio sisältää myös joitain tärkeitä tiedostoja. Yksi näistä on tietokannan tauluja kuvaava *schema.rb*-luokka, jota tutkimalla myös kehittäjä voi saada paremman yleiskuvan sovelluksen käyttämän tietokannan rakenteesta. Toinen tärkeä tiedosto on tietokantaan alustavaa dataa luova *seeds.rb*-luokka. Etenkin testiympäristöissä alustavan datan automaattinen luominen on erittäin hyödyllistä, sillä sovelluksen käyttäminen kokeilumielessä voi olla hankalaa, jos toiminnallinen logiikka on sidottu siihen vahvasti kiinni. Tämän lisäksi *db*-kansio sisältää myös tämän työn edellisessä luvussakin lyhyesti esitetyt tietokannan migraatitiedostot, joiden avulla tietokannan rakenteeseen voidaan suorittaa muutoksia. Migraatitiedostot sijaitsevat *db*-kansion sisäisessä omassa *migrate*-alihakemistossaan.

### 6.3 Versionhallinnassa käytetty puurakenne

Git-versionhallintaa käyttämällä on mahdollista luoda joustava rakenne kehitystyön tekemiseen, etenkin jos kehittäjiä on enemmän kuin yksi. Fluido Portalin kehitystyössä tätä hyödynnettiin luomalla erillinen sovellusvarasto (*repository*) Fluidon jo ennestään käyttämään *BitBucket*-palveluun, joka tarjoaa myös ilmaisia yksityisiä sovellusvarastoja



käyttäjilleen. Vaihtoehtoisesti olisi ollut mahdollista käyttää myös muita sovellusvarastopalveluita, kuten *GitHubia*, mutta sen tarjoamat yksityiset sovellusvarastot ovat maksullisia. *BitBucket*-palvelun ohella myös Herokussa sijaitsee yksi Fluido Portalin sovellusvarastoista.



Kuva 10: Fluido Portalin käyttämät Git-versionhallinnan sovellusvarastot ja niiden haarat.

Fluido Portalin versionhallinnan kokonaisrakenteesta (kuva 10) voidaan huomata *BitBucket*-palvelun sovellusvarastossa sijaitsevat neljä haaraa (*branch*), joissa varsinainen kehitystyö on suoritettu. Haarat *dev-totte* ja *dev-aleksi* vastaavat kahden eri kehittäjän omia kehityshaaroja, joissa on toteutettu erilaisia toiminnallisuuksia Fluido Portalisiin. Kun jokin toiminnallisuus on valmistunut, se on siirretty *integration*-haaraan, jossa kehittäjien muutokset yhdistetään toimivaksi paketiksi. Kun yhdistäminen valmistuu, voidaan *master*-haara, jonka tarkoituksena on säilyttää viimeisin toimiva versio sovelluksesta, päivittää. Eri haarojen kanssa työskentely on Gitin ansiosta suhteellisen helppoa ja turvallista, sillä kehittäjän on mahdollista ottaa haluamansa haara käyttöön ja yhdistää havaitut muutokset turvallisesti toisen haaran kanssa. Git tutkii aina tiedostojen sisällön perusteella, SHA1-tiivistefunktiota käyttäen, onko niissä tapahtunut muutoksia [11].

Uuden Heroku-sovelluksen luonnin yhteydessä sille luodaan automaattisesti oma sovellusvarasto, jossa julkaistua sovellusta säilytetään. Koska kehitystyötä on tehty Herokun ul-

kopuolisessa Git-sovellusvarastossa, ei Herokuun ole tarvittu erillisiä lisähaaroja sen oman *master*-päähaaran lisäksi. Herokun päähaaraan on eri kehitysvaiheissa aika-ajoin päivitetty jokin *BitBucket*-sovellusvaraston haaroista, mutta hyvänä tavoitteena on säilyttää yhdenmukaisuus molempien *master*-haarojen välillä.

## 6.4 Käyttöliittymän esittely

Fluido Portalin käyttöliittymän tarjoamat toiminnallisuudet ovat tässä kehitysvaiheessa vielä hieman suppeat, mutta asiakkaiden projekteihin liittyvät näkymät ovat pitkälti kehitetty valmiiksi. Käyttöliittymän rakentamisessa on hyödynnetty Bootstrapin tarjoamia valmiita CSS-tyylejä ja suunnitteluohjeita.

Fluido Portal	Community	Projects	Support
---------------	-----------	----------	---------

Demo Company /	<b>Projects</b>
----------------	-----------------

Project Name	Actual Hours	Progress	Status
<a href="#">Monster Cat Finland - AutoSync (Heroku Scheduler)</a>	24.0	80.0	Build
<a href="#">Monster Cat Finland - Cat Video Integration</a>	27.0	11.0	Design ongoing

Copyright © 2014 FLUIDO LTD.   Salesforce CRM-Ratkaisut ja Konsultointi   Tekniikantie 14, 02150 Espoo, Finland   +358 6 781 2121
-----------------------------------------------------------------------------------------------------------------------------------

Kuva 11: Projektien listaus Projects-näkymässä.

Yllä olevassa kuvassa 11 esitellään Fluido Portalin *Projects*-näkymä, joka toimii tällä hetkellä käyttäjän ensimmäisenä näkymänä. Siinä listataan taulumaisesti asiakkaan kaikki projektit ja niiden tärkeimmiksi luokitellut tiedot. Listauksen *Project Name* -kolumnissa on linkki kyseisen projektin tarkempaan tarkasteluun johtavaan näkymään.

Fluido Portal
Community
Projects
Support

Project / **Monster Cat Finland - Cat Video Integration**

**Stage:**
Design ongoing

**Actual Hours:**
27.0

**Progress:**

**Work Hours**

Resource	Category	Notes	Hours
Totte Tamminen	Meeting	jippii	3.0
Totte Tamminen	RFC/BR	Blaa blaa	7.0
Aleksi Rajakoski	Project Management	Listening To Abbas Discography From Spotify and Dancing	6.5
Aleksi Rajakoski	Project Management	Just Slacking Off - It's fun	4.0
Aleksi Rajakoski	Project Management	I like Youtube	1.5
Aleksi Rajakoski	Project Management	Morning routines.	3.0
Aleksi Rajakoski	Project Management	Googel LolCatZ	2.0

Back to Projects

Copyright © 2014 FLUIDO LTD. | Salesforce CRM-Ratkaisut ja Konsultointi | Tekniikantie 14, 02150 Espoo, Finland | +358 6 781 2121

Kuva 12: Project-näkymä.

Projektin tarkemmat tiedot esittävä *Project*-näkymä (kuva 12) sisältää yleisten tietojen lisäksi taulumaisen listauksen (kuvassa *Work Hours*) tuntikirjauksista ja niiden tiedoista. Project-näkymän kuvaa tarkasteltaessa voidaan huomata myös CSS-tyyleistä aiheutuva käyttöliittymän graafinen virhe, jossa projektin etenemistä kuvaava palkki ei toimi oikein. Tämä johtuu vielä tuntemattomasta syystä, joka ilmenee vain, kun sovellusta käytetään Salesforce.comin sisällä.

Fluido Portalin hyödyntämä Bootstrap mahdollistaa responsiivisten näkymien helpon kehittämisen, jonka avulla erikokoisia näyttöjä käytettäessä verkkosivuston ulkoasu skaalautuu niihin automaattisesti. Tämän ansiosta käyttäjän ei tarvitse rullata sivua sivusuuntiin esimerkiksi silloin, kun käytössä on jokin pieni-resoluutioinen mobiililaitte.

## Demo Company / Projects

Project Name	Actual Hours	Progress	Status
Monster Cat Finland - AutoSync (Heroku Scheduler)	24.0	80.0	Build
Monster Cat Finland - Cat Video Integration	27.0	11.0	Design ongoing

Copyright © 2014 FLUIDO LTD. | Salesforce CRM-Ratkaisut ja Konsultointi | Tekniikantie 14, 02150 Espoo, Finland | +358 6 781 2121

Kuva 13: Projects-näkymän responsiivinen ulkoasu.

Pienennetty *Projects*-näkyvä (kuva 13) muuttaa ulkoasuansa responsiivisuutensa myötä niin, että taulumaiset listaukset kapenevat ja tekstit rivittyvät uusille riveille. Yksi merkittävimmistä muutoksista on myös päänavigointipalkin linkkien siirtyminen piiloon siihen ilmestyneen painikkeen alle.

### 6.5 Jatkokehityssuunnitelmat

Fluido Portalin tarjoama toiminnallisuus projektien seuraamiseen on ensimmäinen askel sen kehitystyössä. Jotta sovelluksesta saataisiin enemmän irti, tarvitaan myös muita hyödyllisiä toiminnallisuuksia antamaan sille lisäarvoa, joista asiakkaat ja Fluido hyötyvät. Suunnitelmissa on mahdollisesti muun muassa asiakasyhteisön kehittäminen, joka toimisi yhteisenä keskustelukanavana eri asiakkaiden välillä. Tämän lisäksi mahdollisesti kehitettävä tukiyhteisö tarjoaisi asiakkaan ja Fluidon välille keskitetyn kanavan tukipyynnöille,

kysymyksille ja yleiseen keskusteluun. Myös Force.com Canvasin tarjoamien vuorovaikutteisten ominaisuuksien hyödyntämisen käyttöastetta tulisi kasvattaa, jotta sen maksimaalinen hyöty voitaisiin ottaa irti. Tällä hetkellä vuorovaikutteisia toiminnallisuuksia ei ole hyödynnetty juuri ollenkaan, jos niiden määrää verrataan niihin toiminnallisuuksiin, jotka Force.com Canvas mahdollistaisi.

Uusien toiminnallisuuksien lisäksi Fluido Portalin kehityssunnitelmassa on Fluido Portal Synchronizerin vaihtaminen Salesforce.comin tarjoamaan *CloudConnect*-sovellukseen, joka vaihdon jälkeen vastaisi tietokantojen synkronoinnista edistyneemmin keinoin Herokun ja Fluidon Salesforce.com-instanssin välillä. Sen avulla myös tarvittavien tietokantaoperaatioiden määrä kutistuu, jolloin myös riski käyttörajoitteiden saavuttamisesta Herokun ja Salesforce.comin puolella laskee. Näkymien ulkoasun ja käytettävyyden parantaminen on myös tarvittava kehitysaskel ennen Fluido Portalin virallista julkaisua, jotta asiakkaiden vaikutelma sen laadukkuudesta olisi parempi.

Herokun maksullisten ominaisuuksien tilaaminen voi myös olla tarpellista, sillä niiden avulla pystyttäisiin poistamaan sovelluksien uudelleenkäynnistymisestä aiheutuva viive, joka on seuraus sovelluksen käyttämättömyyden aiheuttamasta horrostilasta. Viive voidaan poistaa toisen Herokun tarjoaman virtuaalisen laskentayksikön (*Dynon*) käyttöön-oton myötä, joka antaisi sovellukselle samalla myös lisää virtuaalista prosessointitehoa. Myös Herokun tarjoama maksullinen versio PostgreSQL-tietokannasta voi olla tarpeen, jos lopullinen julkaistavan version tarvitsema datamäärä ylittää ilmaisversiolle asetetun kymmenentuhannen rivin rajoitteen. Julkaisun jälkeen on myös mahdollista sisällyttää Fluido Portal muihin kustomoituihin Salesforce.com-sovelluksiin, jolloin se toimisi lisäosana niiden tarjoamissa kokonaisuuksissa.

Jatkossa Fluido Portalin kehitystyöhön on osallistumassa useampia Fluido Oy:n työntekijöitä, jotka vastaisivat uusien toiminnallisuuksien kehittämisestä sovellukseen avoimella aikataululla. Lopullinen julkaisuaikajankohta Fluido Portalille on vielä auki, eikä sille ole asetettu erityisiä vaatimuksia sen suhteen.

## 7 Yhteenveto ja päätelmät

Tämän insinööritöön tavoitteena oli kehittää Fluido Oy:n asiakkaita varten Ruby on Rails -pohjainen Fluido Portal -niminen sovellus, jonka avulla asiakkaat voivat selata heidän projektejensa etenemistä. Sovellusta käytetään Salesforce.comin sisällä Force.com Canvas -sovelluksena, jota suoritetaan todellisuudessa Heroku-pilvipalvelualustan päällä.

Fluido Portal ei ole valmiusasteeltaan vielä täysin julkaistavassa kunnossa, vaikka suurin osa siihen alustavasti suunnitelluista toiminnallisuuksista on jo toiminnassa. Sitä voidaan jo kuitenkin käyttää esimerkkinä Force.com Canvasin avulla toteutettujen kolmannen osapuolen sovellusten ja Salesforce.comin yhteistoiminnasta. Fluido Portalin tässä työssä esitelty versio tarjoaa hyvän pohjan jatkokehitykselle, joka sisältää muun muassa lisätoiminnallisuuksien toteuttamista ja ulkoasun hiomista laadukkaammalle tasolle, sekä sen mahdollista integroimista muihin kustomoituihin sovelluksiin Salesforce.comin sisällä. Kokonaisuudessaan Fluido Portalin kehitystyö on onnistunut insinööritöitä varten asetettujen tavoitteiden mukaisesti hyvin.

Ruby-ohjelmoinnin osalta on huomionarvoista, kuinka nopeasti on mahdollista toteuttaa toimivia järjestelmiä käyttäen Ruby on Railsia ja muita Gemejä. Yksinkertaisen asiakasportaalin kehittäminen ei ollut mahdoton tehtävä edes hyvin minimaalisella kokemuksella. Rubyssa on myös selkeästi mietitty, miten kehittäjien työtä saadaan helpotettua ja nopeutettua. Sen kielellinen asu on lyhyttä ja ytimekästä, sekä eri tietokantajärjestelmien käyttäminen on suoraviivaista ja syntaksiltaan yhtenäistettyä. Myös valmiiden toiminnallisuuksien (Gemien) käyttöönotto ja hallinta on helppoa, kuten myös käytettävän Ruby-version valinta ja asentaminenkin.

Fluido Portalin tietokantaratkaisuna toimiva Herokun PostgreSQL ja sen synkronointi Fluidon Salesforce.com-instanssin välillä toteutettiin erillisellä Ruby on Rails -sovelluksella, joka kuitenkin jakaa pääsovelluksen kanssa saman tietokannan. Koska synkronointisovellus on eriytetty, voidaan se myös tarvittaessa vaihtaa parempaan vaihtoehtoon, kuten Salesforce.comin tarjoamaan *CloudConnect*-sovellukseen. Näin ollen Fluido Portalin koodikantaa ei jouduta erikseen siivoamaan, jos vaihto CloudConnectiin päätetään myöhem-

mässä vaiheessa toteuttaa. Tällä hetkellä synkronointisovellus toimii automaattisesti Herokussa ajastetun toiminnon avulla, mutta tarvittaessa synkronointi voidaan toteuttaa myös manuaalisesti sen käyttöliittymästä käsin. Mahdollinen tarve synkronointisovelluksen korvaamiseen muodostuu sen yksinkertaisuudesta, joka ei pysty kilpailemaan CloudConnectin tarjoamien ominaisuuksien kanssa.

Heroku on osoittautunut erittäin käteväksi ja helppokäyttöiseksi pilvipalvelualustaksi, jonka avulla kehittäjien omia sovelluksia voidaan julkaista. Maksutta suoritettavissa sovelluksissa on tiettyjä rajoituksia, mutta tarvittaessa niihin voidaan ostaa kuukausiveloituksella käyttöön erilaisia ominaisuuksia. Yksi tärkeä syy maksullisen lisäpalvelun ostamiseen voi olla Heroku-sovellusten käyttämien *Dynojen* määrän kasvattaminen. Oletuksena sovelluksilla on käytössään vain yksi *Dyno*, joka aiheuttaa osaltaan huomattavankin hitaita käynnistymisaikoja sovellusta ladattaessa. Tämä johtuu siitä, että sovellukset sammutetaan tietyn ajanjakson jälkeen kun niitä ei ole käytetty hetkeen. Toisen *Dynon* käyttöönotto estää sovelluksen sammumisen, antaen sille myös lisää suorituskykyä. Sovelluksen lepo-tilan jälkeinen käynnistyminen aiheutti aina harmittavia lyhyitä viiveitä Fluido Portalin kehitystyötä tehtäessä, jonka vuoksi sen julkaisuajankohtana toisen *Dynon* tilaaminen olisi perusteltua, jotta asiakkaat saisivat paremman käyttökokemuksen. Heroku-sovellusten laajennettavien ominaisuuksien joukossa on myös ilmaiseksi käytössä olevia tietokantoja, kuten Fluido Portalissa käytettävä PostgreSQL. Riippuen lopullisesta tallennettavan datan kokonaismäärästä tietokannan päivittämistä maksulliseen versioon kannattaa myös harkita. Maksullisen version myötä tietokannassa on enemmän tilaa datan tallennusta varten, eikä sen päivittäminen aiheuta erillisiä toimenpiteitä.

## Lähteet

- 1      Salesforce.com. Wikipedia; 2014. Saatavilla osoitteesta:  
<http://en.wikipedia.org/wiki/Salesforce.com> [viitattu 2.3.2014].
- 2      Salesforce.com. Fluido Oy; 2014. Saatavilla osoitteesta:  
<http://www.fluido.fi/salesforce.html> [viitattu 2.3.2014].
- 3      Ruby (programming language). Wikipedia; 2014. Saatavilla osoitteesta:  
[http://en.wikipedia.org/wiki/Ruby\\_%28programming\\_language%29](http://en.wikipedia.org/wiki/Ruby_%28programming_language%29)  
[viitattu 22.3.2014].
- 4      Ruby On Rails. Wikipedia; 2014. Saatavilla osoitteesta:  
[http://fi.wikipedia.org/wiki/Ruby\\_on\\_Rails](http://fi.wikipedia.org/wiki/Ruby_on_Rails) [viitattu 29.1.2014].
- 5      PostgreSQL. Wikipedia; 2014. Saatavilla osoitteesta:  
<http://fi.wikipedia.org/wiki/PostgreSQL> [viitattu 16.1.2014].
- 6      Bootstrap (front-end framework). Wikipedia; 2014. Saatavilla osoitteesta:  
[http://en.wikipedia.org/wiki/Bootstrap\\_\(front-end\\_framework\)](http://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))  
[viitattu 2.3.2014].
- 7      Heroku. Wikipedia; 2014. Saatavilla osoitteesta:  
<http://en.wikipedia.org/wiki/Heroku> [viitattu 22.3.2014].
- 8      Force.com Canvas (Pilot). Salesforce.com; 2013. Saatavilla osoitteesta:  
[http://developer.force.com/releases/release/Winter13/Forcedotcom+](http://developer.force.com/releases/release/Winter13/Forcedotcom+Canvas+Pilot)  
[Canvas+Pilot](http://developer.force.com/releases/release/Winter13/Forcedotcom+Canvas+Pilot) [viitattu 22.3.2014].
- 9      Git. Wikipedia; 2014. Saatavilla osoitteesta:  
<http://fi.wikipedia.org/wiki/Git> [viitattu 22.3.2014].
- 10     Thomas, Hansson. Agile Web Development with Rails. Yhdysvallat: The Pragmatic Programmers LLC; 2005.
- 11     Loeliger, McCullough. Version Control with Git, Second Edition. Yhdysvallat: O'Reilly Media, Inc; 2012.